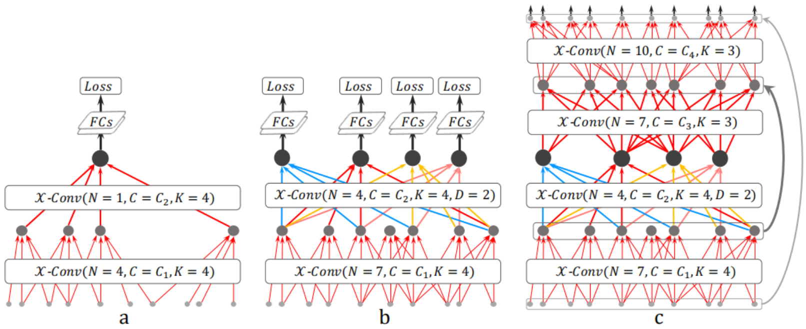

Figure 4: PointCNN architecture for classification (a and b) and segmentation (c), where N and C denote the output representative point number and feature dimensionality, K is the neighboring point number for each representative point, and D is the $\mathcal{X}$-Conv dilation rate.

图 1: PointCNN

X-transformation:

- 给点的特征进行赋权重
- 将点转置到一个潜在的标准顺序

ALGORITHM 1: $\mathcal{X}$-Conv Operator	
Input :$\mathbf{K}, p, \mathbf{P}, \mathbf{F}$	
Output :$\mathbf{F}_p$	▷ Features "projected", or "aggregated", into representative point p
1: $\mathbf{P}' \leftarrow \mathbf{P} - p$	▷ Move $\mathbf{P}$ to local coordinate system of p
2: $\mathbf{F}_\delta \leftarrow MLP_\delta(\mathbf{P}')$	▷ **Individually** lift each point into C_δ dimensional space
3: $\mathbf{F}_* \leftarrow [\mathbf{F}_\delta, \mathbf{F}]$	▷ Concatenate $\mathbf{F}_\delta$ and $\mathbf{F}$, $\mathbf{F}_*$ is a $K \times (C_\delta + C_1)$ matrix
4: $\mathcal{X} \leftarrow MLP(\mathbf{P}')$	▷ Learn the $K \times K$ $\mathcal{X}$-transformation matrix
5: $\mathbf{F}_\mathcal{X} \leftarrow \mathcal{X} \times \mathbf{F}_*$	▷ Weight and permute $\mathbf{F}_*$ with the learnt $\mathcal{X}$
6: $\mathbf{F}_p \leftarrow \text{Conv}(\mathbf{K}, \mathbf{F}_\mathcal{X})$	▷ Finally, typical convolution between $\mathbf{K}$ and $\mathbf{F}_\mathcal{X}$

图 2: X-conv

$$\mathbf{F}_p = \mathcal{X} - \text{Conv}(\mathbf{K}, p, \mathbf{P}, \mathbf{F}) = \text{Conv}\left(\mathbf{K}, MLP(\mathbf{P} - p) \times [MLP_\delta(\mathbf{P} - p), \mathbf{F}]\right)$$

分类任务 有两种网络，分别对应图 1 的 a,b。b 在保留相同卷积深度的基础上，加入了卷积 dilation，增加了感受域，但未卷积核的大小。

语义分割 在特征提取之后，进行" 反卷积"，还原回最初的点的个数。